This listing of claims will replace all prior versions, and listings, of claims in the application:

LISTING OF CLAIMS

1.   (Original) A method of specifying an asynchronous web service within a procedural programming environment, the method comprising:

        providing a source code representation of at least a portion of web service logic, the logic including at least one method declared to be a callback method; and

        identifying a member variable "**a source code representation**" for interacting asynchronously with the client using said callback method, and to assign the client proxy object to said member variable.

2.   (Original) The method of claim 1, wherein said callback method is declared inside a callback interface definition.

3.   (Original) The method of claim 1, further comprising:

        specifying one or more declarative annotations associated with said callback method to cause a compiler to generate one or more persistent components to maintain conversational state related to the identified member variable.

4.   (Original) The method of claim 3, wherein the one or more declarative annotations indicate to the compiler whether the identified method is at least one of a start method, a continue method, and a finish method, wherein the start method applies to the start of a

SEA/109870/130103/KHF/334710.1

stateful conversation between the client and the web service, the continue method applies to the continuation of an ongoing stateful conversation between the client and the web service, and the finish method applies to the completion of an ongoing stateful conversation between the client and the web service.

5.   (Original) The method of claim 4, wherein when a method declared to be a start method is invoked at run-time, a new instance of a conversation is created, and a unique identifier is associated with that conversational instance to facilitate management of multiple simultaneous conversations.

6.   (Original) The method of claim 4, wherein when a method declared to be a continue method or a finish method is invoked at run-time, a unique identifier is obtained and used to access a corresponding instance of a conversation.

7.   (Original) The method of claim 6, wherein the unique identifier is provided by the client.

8.   (Original) The method of claim 6, wherein when a method declared to be a finish method is invoked at run-time, the corresponding instance of the conversation is destroyed after processing by the web service logic.

9.   (Original) The method of claim 3, wherein the one or more declarative annotations indicate to the compiler whether the identified callback method is buffered, causing the

compiler to instantiate one or more queues to temporarily store one or more asynchronous responses for delivery to the client when the client is able to receive the responses.

10.   (Original) In a procedural programming environment, a method of returning an asynchronous response to a client, the method comprising:

      identifying a callback interface associated with the client;

      identifying a member variable declared to implement the callback interface;

      generating a proxy object that implements the callback interface; and

      assigning the proxy object to the member variable.

11.   (Original) The method of claim 10, wherein one or more declarative annotations inside the callback interface cause the compiler to generate one or more persistent components to maintain conversational state related to the identified member variable.

12.   (Currently Amended) In an asynchronous web service, a method comprising:

      receiving <u>on one or more computing devices</u> a message from a client requesting that a web service method be invoked;

      parsing <u>on one or more computing devices</u> the message to identify the requested web service method in addition to a callback address indicating a location where the client is listening for callbacks from the web service;

      storing <u>on one or more computing devices</u> the callback address in association with a previously generated proxy object; and

      invoking <u>on one or more computing devices</u> the requested web service method.

13.   (Currently Amended) The method of claim 12, further comprising:

identifying on one or more computing devices an instance identifier provided by the client indicating a particular instance of the client that is listening for callbacks from the web service; and

storing on one or more computing devices the instance identifier in association with the callback address and the proxy object.

14.   (Original) The method of claim 13, wherein at least one of the callback address and the instance identifier is encapsulated in one or more request messages.

15.   (Original) The method of claim 13, wherein the instance identifier is embedded within a conversation identifier used to access a corresponding instance of a conversation between the client and the web service.

16.   (Original) The method of claim 14, wherein at least one of the callback address and the instance identifier is encapsulated in one or more SOAP message headers.

17.   (Original) The method of claim 13, wherein the instance identifier is a GUID.

18.   (Currently Amended) The method of claim 12, wherein invoking the requested web service method further comprises:

extracting on one or more computing devices data representation language elements from the message;

mapping <u>on one or more computing devices </u>the data representation language elements to programming language objects;

invoking <u>on one or more computing devices </u>the requested web service method including passing the programming language objects as parameters to the web service method;

mapping <u>on one or more computing devices </u>programming language objects returned by the web service method onto representative data representation language elements;

packaging <u>on one or more computing devices </u>the representative data representation language elements according to one or more protocols used by the client in generating the requested message; and

transmitting<u> on one or more computing devices </u>the packaged data representation elements to the client in accordance with one or more protocols used by the client in transmitting the message to the web service.

19.   (Currently Amended) The method of claim 13, further comprising:

generating <u>on one or more computing devices </u>an asynchronous response to the client in response to the message by invoking a method on the previously generated proxy object using a declared member variable, wherein invoking includes

passing <u>on one or more computing devices </u>programming language objects as parameters to said method,

mapping <u>on one or more computing devices </u>the programming language objects and method invocation onto representative data representation language elements;

packaging on one or more computing devices the representative data representation language elements, and callback instance identifier into an asynchronous response message, and

transmitting on one or more computing devices -the asynchronous response message to the client at the callback address in accordance with one or more protocols used by the client in transmitting the original request message to the web service.

20.   (Original) A method for specifying logic within a procedural programming environment for receiving a callback from an asynchronous web service, the method comprising:

identifying a member variable to be used for interacting with said asynchronous web service; and

providing a method associated with said member variable, the method having a signature and containing logic for receiving said callback from said asynchronous web service.

21.   (Original) The method of claim 20, wherein said method is manually provided by a developer.

22.   (Original) The method of claim 20, wherein the method signature is provided by an integrated development environment based on a specified service description file containing a declaration for said callback.

23.   (Original) The method of claim 20, wherein said method is associated with said member variable using a method naming convention that utilizes the name of said member variable and the name of said callback.

24.   (Original) The method of claim 20, further comprising:

specifying one or more declarative annotations in association with said member variable to cause a compiler to generate one or more persistent components to maintain conversational state related to the asynchronous web service.

25.   (Original) The method of claim 24, wherein the one or more declarative annotations are specified within the source code.

26.   (Original) The method of claim 24, wherein the one or more declarative annotations are specified outside of the source code.

27.   (Original) The method of claim 24, wherein the one or more declarative annotations are manually specified by a developer.

28.   (Original) The method of claim 24, wherein the one or more declarative annotations are automatically specified by an integrated development environment based upon input provided by a developer.

29.   (Original) The method of claim 20, wherein asynchronous responses from the external web service are passed to said method associated with said member variable.

30.   (Currently Amended) In a web service, a method comprising:

generating on one or more computing devices a request to an external web service using a proxy object previously generated by a compiler based upon a service description file associated with the external web service, wherein the request includes a callback address to identify a location to which the external web service should return a response;

transmitting on one or more computing devices the request as a request message to the external web service using one or more transmission protocols; and

receiving on one or more computing devices an asynchronous response from the external web service.

31.   (Original) The method of claim 30, wherein the callback address includes proxy object identifier.

32.   (Original) The method of claim 30, wherein the callback address is included within one or more headers of the request message.

33.   (Original) The method of claim 32, wherein the request message is a SOAP based message.

34.   (Original) The method of claim 30, wherein the callback address comprises a URL identifying a location where the web service is listening for a response from the external web service.

35.   (Original) The method of claim 30, wherein the request further includes a callback instance identifier representing a specific instance of the requesting web service to which asynchronous responses are to be routed.

36.   (Currently Amended) The method of claim 35, wherein ~~and~~ the callback instance identifier is included within one or more headers of the request message.

37.   (Original) The method of claim 36, wherein the request message is a SOAP based message.

38.   (Currently Amended) In a web service, a method comprising:

        receiving <u>on one or more computing devices</u> a message identifying a callback address, a callback method and a proxy object identifier;

        extracting <u>on one or more computing devices</u> the proxy object identifier from the message;

        determining <u>on one or more computing devices</u> a method to be invoked based at least in part upon the proxy object identifier and the callback method; and

        routing <u>on one or more computing devices</u> the request to the method to be invoked.

39.   (Original) The method of claim 38, wherein the callback address comprises a URL indicating a location where the web service listens for callbacks from the external web service.

40.   (Original) The method of claim 38, wherein determining the method to be invoked comprises appending a name associated with the callback method to the proxy object identifier.

41.   (Original) The method of claim 38, wherein the message further identifies a callback instance identifier.

42.   (Currently Amended) The method of claim 41, wherein routing further comprises:

        identifying on one or more computing devices a callback instance based at least in part upon the callback instance identifier; and

        routing on one or more computing devices the request to a method associated with the identified callback instance.

43.   (Original) An article of manufacture comprising:

        a storage medium having stored therein a plurality of programming instructions, which when executed provide a graphical interface to facilitate specification of an asynchronous web service within a procedural programming environment including

        providing a source code representation of at least a portion of web service logic, the logic including at least one method declared to be a callback method, and

identifying a member variable declared to implement said callback method to cause a compiler to generate a client proxy object for interacting asynchronously with the client using said callback method, and to assign the client proxy object to said member variable.

44. (Original) The article of claim 43, wherein said callback method is declared inside a callback interface definition.

45. The article of claim 43, wherein the programming instructions further facilitate specifying one or more declarative annotations associated with said callback method to cause a compiler to generate one or more persistent components to maintain conversational state related to the identified member variable.

46. (Original) The article of claim 45, wherein the one or more declarative annotations indicate to the compiler whether the identified method is at least one of a start method, a continue method, and a finish method, wherein the start method applies to the start of a stateful conversation between the client and the web service, the continue method applies to the continuation of an ongoing stateful conversation between the client and the web service, and the finish method applies to the completion of an ongoing stateful conversation between the client and the web service.

47. (Original) The article of claim 46, wherein when a method declared to be a start method is invoked at run-time, a new instance of a conversation is created, and a unique

identifier is associated with that conversational instance to facilitate management of multiple simultaneous conversations.

48.　　(Original) The article of claim 46, wherein when a method declared to be a continue method or a finish method is invoked at run-time, a unique identifier is obtained and used to access a corresponding instance of a conversation.

49.　　(Original) The article of claim 48, wherein the unique identifier is provided by the client.

50.　　(Original) The article of claim 48, wherein when a method declared to be a finish method is invoked at run-time, the corresponding instance of the conversation is destroyed after processing by the web service logic.

51.　　(Original) The article of claim 45, wherein the one or more declarative annotations indicate to the compiler whether the identified callback method is buffered, causing the compiler to instantiate one or more queues to temporarily store one or more asynchronous responses for delivery to the client when the client is able to receive the responses.

52.　　(Original) An article of manufacture comprising:

　　　　a storage medium having stored therein a plurality of programming instructions designed to program an apparatus to return an asynchronous response to a client, which programming instructions when executed enable the apparatus to

identify a callback interface associated with the client;

identify a member variable declared to implement the callback interface;

generate a proxy object that implements the callback interface; and

assign the proxy object to the member variable.

53.   (Original) The article of claim 52, wherein one or more declarative annotations inside the callback interface cause the compiler to generate one or more persistent components to maintain conversational state related to the identified member variable.

54.   (Original) An article of manufacture comprising:

a storage medium having stored therein a plurality of programming instructions designed to program an apparatus to implement an asynchronous web service, which programming instructions when executed enable the apparatus to

receive a message from a client requesting that a web service method be invoked;

parse the message to identify the requested web service method in addition to a callback address indicating a location where the client is listening for callbacks from the web service;

store the callback address in association with a previously generated proxy object; and

invoke the requested web service method.

55.   (Currently Amended) The article of claim 54, wherein the programming instructions further enable he the apparatus to

SEA/109870/130103/KHF/334710.1

identify an instance identifier provided by the client indicating a particular instance of the client that is listening for callbacks from the web service; and

~~storeng~~ <u>store</u> the instance identifier in association with the callback address and the proxy object.

56. (Original) The article of claim 55, wherein at least one of the callback address and the instance identifier is encapsulated in one or more request messages.

57. (Original) The article of claim 55, wherein the instance identifier is embedded within a conversation identifier used to access a corresponding instance of a conversation between the client and the web service.

58. (Original) The article of claim 56, wherein at least one of the callback address and the instance identifier is encapsulated in one or more SOAP message headers.

59. (Original) The article of claim 55, wherein the instance identifier is a GUID.

60. (Original) The article of claim 54, wherein the programming instructions to cause the apparatus to invoke the requested web service further cause the apparatus to

extract data representation language elements from the message;

map the data representation language elements to programming language objects;

invoke the requested web service method including passing the programming language objects as parameters to the web service method;

map programming language objects returned by the web service method onto representative data representation language elements;

package the representative data representation language elements according to one or more protocols used by the client in generating the requested message; and

transmit the packaged data representation elements to the client in accordance with one or more protocols used by the client in transmitting the message to the web service.

61.   (Original) The article of claim 55, wherein the programming instructions further enable the apparatus to generate an asynchronous response to the client in response to the message by invoking a method on the previously generated proxy object using a declared member variable, wherein invoking includes

passing programming language objects as parameters to said method,

mapping the programming language objects and method invocation onto representative data representation language elements;

packaging the representative data representation language elements, and callback instance identifier into an asynchronous response message, and

transmitting the asynchronous response message to the client at the callback address in accordance with one or more protocols used by the client in transmitting the original request message to the web service.

62.   (Original) An article of manufacture comprising:

a storage medium having stored therein a plurality of programming instructions, which when executed provide a graphical interface to facilitate specification of an asynchronous web service within a procedural programming environment including

identifying a member variable to be used for interacting with said asynchronous web service; and

providing a method associated with said member variable, the method having a signature and containing logic for receiving said callback from said asynchronous web service.

63.   (Original) The article of claim 62, wherein said method is manually provided by a developer.

64.   (Original) The article of claim 62, wherein the method signature is provided by an integrated development environment based on a specified service description file containing a declaration for said callback.

65.   (Original) The article of claim 62, wherein said method is associated with said member variable using a method naming convention that utilizes the name of said member variable and the name of said callback.

66.   (Original) The article of claim 62, wherein the programming instructions further facilitate

specifying one or more declarative annotations in association with said member variable to cause a compiler to generate one or more persistent components to maintain conversational state related to the asynchronous web service.

67.   (Original) The article of claim 66, wherein the one or more declarative annotations are specified within the source code.

68.   (Original) The article of claim 66, wherein the one or more declarative annotations are specified outside of the source code.

69.   (Original) The article of claim 66, wherein the one or more declarative annotations are manually specified by a developer.

70.   (Original) The article of claim 66, wherein the one or more declarative annotations are automatically specified by an integrated development environment based upon input provided by a developer.

71.   (Original) The article of claim 62, wherein asynchronous responses from the external web service are passed to said method associated with said member variable.

72.   (Original) An article of manufacture comprising:

a storage medium having stored therein a plurality of programming instructions designed to program an apparatus to implement an asynchronous web service, which when

executed enable the apparatus to generate a request to an external web service using a proxy object previously generated by a compiler based upon a service description file associated with the external web service, wherein the request includes a callback address to identify a location to which the external web service should return a response;

transmit the request as a request message to the external web service using one or more transmission protocols; and

receive an asynchronous response from the external web service.

73. (Original) The article of claim 72, wherein the callback address includes proxy object identifier.

74. (Original) The article of claim 72, wherein the callback address is included within one or more headers of the request message.

75. (Original) The article of claim 74, wherein the request message is a SOAP based message.

76. (Original) The article of claim 72, wherein the callback address comprises a URL identifying a location where the web service is listening for a response from the external web service.

77.   (Original) The article of claim 72, wherein the request further includes a callback instance identifier representing a specific instance of the requesting web service to which asynchronous responses are to be routed.

78.   (Original) The article of claim 77, wherein and the callback instance identifier is included within one or more headers of the request message.

79.   (Original) The article of claim 78, wherein the request message is a SOAP based message.

80.   (Original) An article of manufacture comprising:

a storage medium having stored therein a plurality of programming instructions designed to program an apparatus to implement an asynchronous web service, which when executed enable an apparatus to

receive a message identifying a callback address, a callback method and a proxy object identifier;

extract the proxy object identifier from the message;

determine a method to be invoked based at least in part upon the proxy object identifier and the callback method; and

route the request to the method to be invoked.

81.   (Original) The article of claim 80, wherein the callback address comprises a URL indicating a location where the web service listens for callbacks from the external web service.

82.   (Original) The article of claim 80, wherein the programming instructions to enable the apparatus to determine the method to be invoked further comprises programming instructions to enable the apparatus to append a name associated with the callback method to the proxy object identifier.

83.   (Original) The article of claim 80, wherein the message further identifies a callback instance identifier.

84.   (Original) The article of claim 83, wherein the programming instructions to enable the apparatus to route further enable the apparatus to

identify a callback instance based at least in part upon the callback instance identifier; and

route the request to a method associated with the identified callback instance.